C246 Lecture Notes (2/13/2003)

Barbara Engelhardt

February 21, 2003

In this class we discussed how to describe the statistical significance of database search techniques and sequence comparison methods. For this discussion, all log functions without subscripts are the natural logarithm.

1 Comparison of Database Search Techniques

References:

Pearson WR. Comparison of methods for searching protein sequence databases. *Protein Science.* 1995 June; 4(6):1145-60.

Brenner SE, Chothia C, Hubbard TJP. 1998. Assessing sequence comparison methods with reliable structurally-identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences of the United States of America 95*:6073-6078.

Green RE, Brenner SE. 2002 Bootstrapping and normalization for enhanced evaluations of pairwise sequence comparison. *Proceedings of the IEEE 9*:1834-47.

A quick summary of some of the results:

- BLAST using the scoring matrix BL50 works the best. BL50 is a scoring matrix that relies on the assumption that over a short time there will be local differences, or generally that low-level similarities over a longer evolutionary time are less likely than high-level similarities over a short evolutionary time.

- FASTA, SSEARCH perform well with the VTML160 matrix (discussed in the last lecture).

- BLAST-WU had excellent results, but the statistics that it produced were inferior to the other methods.

Generally, 90% of evolutionarily related sequences are never found at all, even with an exact search method. Many of the heuristic methods only lose $2 - 3\%$ more of the evolutionarily related sequences, which is why they are used so frequently.

1.1 Segmentation

SEG, or "segmentation of sequences by local complexity" (also called *masking* or *filtering*), blocks out low-complexity regions of sequences before the comparisons begin. This method is not useful for PAM/BLOSUM matrices, which make relative frequency assumptions uniformly across the length of the proteins. This method is good for understanding the 3 dimensional structure or folds in protein comparisons.

Given a window size L (say 12 positions), for every sequential set of 12 amino acids, compute:

$$K = \frac{1}{L} \log_2 \left(\frac{L!}{\Pi_i n_i!} \right)$$

where K is a measure of the complexity of the region, n_i is a count of the number of residues in that window of type i (for all residues i).

The window is considered low complexity if $K \leq K_{trigger}$, and similarly, the window size L is extended if $K \leq K_{extend}$, where $K_{trigger} \leq K_{extend}$. The regions that are of low complexity then are not used in the comparison methods.

2 Statistics of Sequence Comparison

References:

http://www.people.virginia.edu/ wrp/papers/stat_gen_00.pdf

Stephen Altschul's web page.

The scores for particular "matches" need some context:

- How long is the sequence?

- Which method was used in the search?
- How did you find the similar sequences?
- What is the random chance that this score would appear if random sequences are aligned?

For the final question, it is straightforward to compute the likelihood of a particular score given a distribution over randomly aligned sequences pulled from the same distribution as the sequences that are being compared.

For global alignments, the type of distribution is unknown. One method is to assume a type of distribution and learn the distribution parameters by a bootstrap method: shuffle the sequence and aligning two random shufflings some large number of times, and derive the parameters for the distribution from the sample scores.

For gapped local alignments, it is straightforward to compute $z = \frac{s-\mu}{\theta}$, or the number of standard deviations out from the mean μ a specific score s is, given that you have selected a Gaussian model. But the problem is that the model is not Gaussian. In practice, gapped local alignment is analyzed using ungapped local alignment methods with a few caveats and modifications.

3 Ungapped Local Alignment

Longest common subsequence (LCS) methods: The E-value for a particular score can be characterized by the following general equation:

$$E = \log_{\frac{1}{p}} kmn$$

where p is the total number of characters, k is the weighting, and m, n are the length of the two alignments (not the aligned length, but the full length).

This leads us to consider the extreme value distribution (EVD) to model the distribution of randomly aligned sequences (see equations later for motivation). The statistical interpretation of the Normal distribution is that it is the sum of many IID (independent, identically distributed) random variables. The statistical interpretation of the EVD is that it the maximum over many IID random variables (with an arbitrary distribution). Graphically this means the density curves of the EVD for a particular Gaussian is shifted to the left and has a slower drop off than the Gaussian on the right. The EVD was first introduced by Gumbel to describe the frequency of floods.

The distribution of the EVD (double exponential), both unparameterized and parameterized with u and λ:

$$P(s \leq x) = e^{-e^{-x}}$$

$$P(s \leq x) = e^{-e^{-\lambda(x-u)}}$$

where λ represents the width of the density (corresponding to σ, the standard deviation, in the Gaussian density), and u represents the modal point (or the characteristic value).

There is a quick mapping from Gaussian parameters to EVD parameters:

$$\mu = u + \frac{\gamma}{\lambda}$$

$$\theta = \frac{\pi}{\lambda\sqrt{6}}$$

where $\gamma = \frac{1}{e}$ is Euler's number.

3.1 Analytical Parameters

For ungapped alignments, the parameters for the EVD can be learned analytically as follows (from Karlin-Altschul (Dembo)):

$$\lambda = \{x : \sum_i \sum_j p_i p_j e^{s_{ij}x} = 1, x \geq 0\}$$

$$u = \frac{\log kmn}{\lambda}$$

$$k = f(s_{ij}p_{ij})$$

where $f(\cdot)$ is some function of the score for a particular i, j score and the likelihood of that pair being aligned.

Moreover, they pointed out that the distribution can be estimated as follows:

$$P(s \geq x) = 1 - e^{-Kmne^{-\lambda x}} \sim kmne^{-\lambda x}$$

Notice that λ acts as a scaling factor for a scoring matrix:

$$s_{ij} = \frac{\log\left(\frac{q_{ij}}{p_i p_j}\right)}{\lambda}$$

then k multiplies the search space appropriately.
So then the p-value for some score s is:

$$E(\text{pairs with score } \geq s) = kmne^{-\lambda s}.$$

3.2 Bit Scores

If the score is not in bits, then it can easily be converted to bits (and results from different scoring matrices can be compared, for example):

$$s^* = \frac{\lambda s \log k}{\log 2}$$

$$E = mn2^{-s^*}$$

3.3 Edge Effects

The problem of edge effects is that alignments might not be made over whole sequence (this is the problem that arises in global alignment that no end gap penalties helps to alleviate). Generally, the problem is that m and n are not infinite although the E-value distribution makes that assumption. The method that exists to account for this is to rescale m and n to remove the expected length of a match (in the following, $E(l)$ is the expected length for a match):

$$\hat{m} = m - E(l)$$

$$\hat{n} = n - E(l)$$

$$\hat{E} = k\hat{m}\hat{n}e^{-\lambda s} = \hat{m}\hat{n}2^{-s^*}$$

This method was used in earlier versions of BLAST.

3.4 Complete Alignments

For an ungapped alignment with multiple disjoint high-scoring regions, the regions can be combined with gaps connecting the regions. For this method, two different scoring methods have been proposed: *Poisson statistics* and *sum statistics* (Gish, Altschul).

The Poisson statistics greatly exaggerated the significance of the matches, whereas the sum statistics work better than the Poisson statistics by summing over the aligned regions and adding penalties for the intermediate gaps, but still this was not a clean method of scoring matches of this type.

4 Gapped Alignments

In general, the problem of generating statistics for gapped alignments is hard but possible, although there are no really robust ranking statistics. If the alignment of two sequences is truly local, gapped alignment has an EVD distribution also. How can you derive the parameters λ and k? The previous analytic method is no longer possible since it requires ungapped alignments. There are three empirical methods used to generate the equivalent parameters for gapped alignments:

- *Empirical Simulation*: Generate $10,000$-$1,000,000$ random sequences, with the same frequency distribution as the database to search. Align them pairwise, and find the mean and standard deviation of the score for each of the alignments. Using the mapping above, μ, σ can be converted to λ, u. Notice that there is a single length of original sequence so there is a single set of parameters for that length.

- *Empirical Database Search:* The problem with the above method is that the composition of the samples is fixed, whereas in the real dataset the composition can vary wildly. Instead of randomly generating the sequences, randomly pull them from the database being used. Instead of a single set of parameters, the μ and σ must be found for each pair of lengths for the random sequences being aligned.

- *Shuffle Sequence:* Randomly shuffle the particular sequence, and align the sequence against a shuffling of itself. The obvious problem is that the random shuffle will still have a particular composition of amino acids.

These methods are not used with BLAST1, but are used in and after BLAST2. The parameters are found empirically only after full Smith-Waterman is performed on all of the random pairs of aligned sequences. FASTA uses the empirical database search method and in practice it works well.

Of course these methods are only pairwise. How can we scale up to the whole search?

BLAST calculates the E-value with respect to the length of the entire database, specifically, for N the number of amino acids in a particular database,

$$E_{BLAST} = kNne^{-\lambda s}$$

where s is the score in bits of the particular alignment.

FASTA does a similar scaling, but with respect to D, the number of sequences present in the comparison database:

$$E_{FASTA} = kDnme^{-\lambda s}$$

If the average length of the sequences in the database is similar, then the results of these two ranking methods is very similar. In general, FASTA cares how big the matching sequence is: the longer the match sequence, the less significant the match appears. The problem with this assumption in practice is that the dependency on length does not make sense for genomic data, since length should not be a factor in the significance of the match.

5 How are the E-scores used in each of the heuristic search algorithms?

- **BLAST1:** Found an analytic k, λ, and s_{ij} from the user's chosen score matrix.

- **BLAST2:** Found k, λ from empirical simulations. Requires that score matrix be one which has an associated empirically found k, λ.

- **BLAST2.2.2**: Similar to BLAST2 but with an additional tweak: computed how far empirical k, λ was from the ungapped k, λ, found using empirical simulations with same score matrix. The new parameters are tweaked to account for this discrepancy in the original k, λ, without any real theoretical justification.

- **PSI-BLAST:** Generates the score matrix on the fly for a fixed k, λ. Specifically, each s_{ij} is chosen in an attempt to achieve the fixed k, λ EVD parameters.

- **C-WA?**: (UCSD) Generates k, λ from empirical database search using a database smaller than the database for the total search.

- **FASTA:** Chooses k, λ based on an empirical database search.